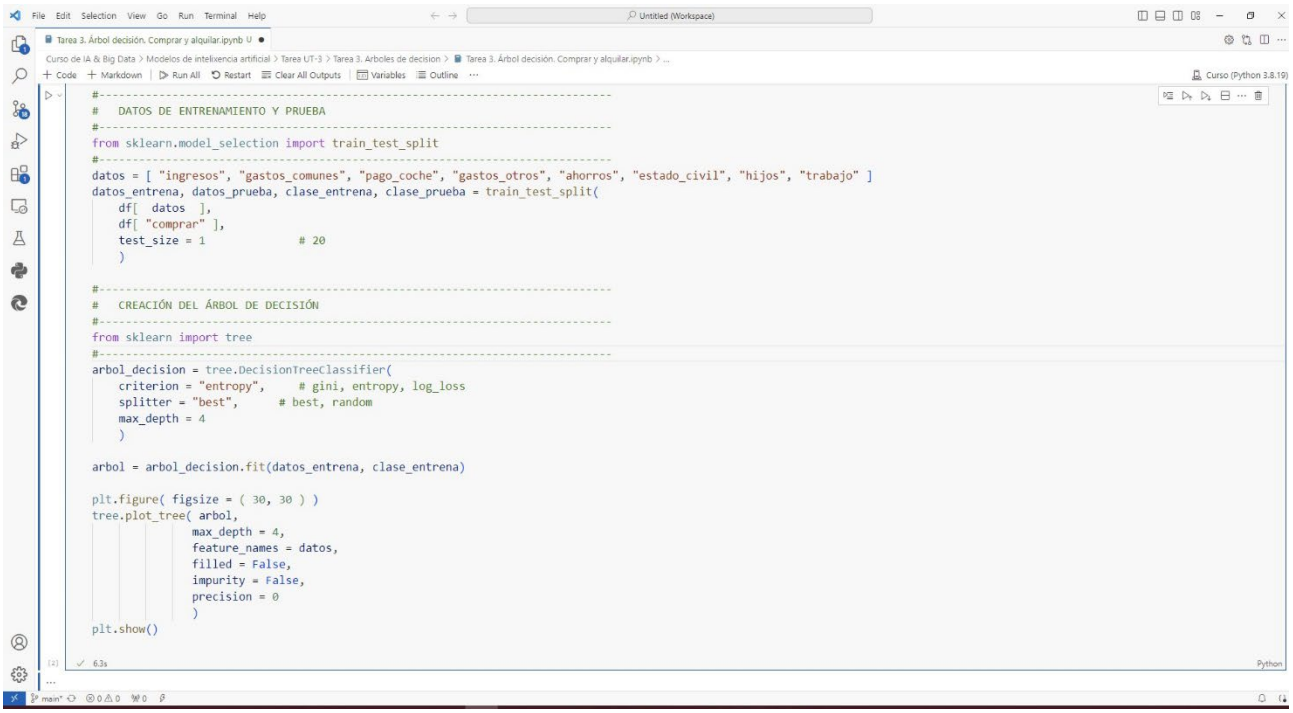
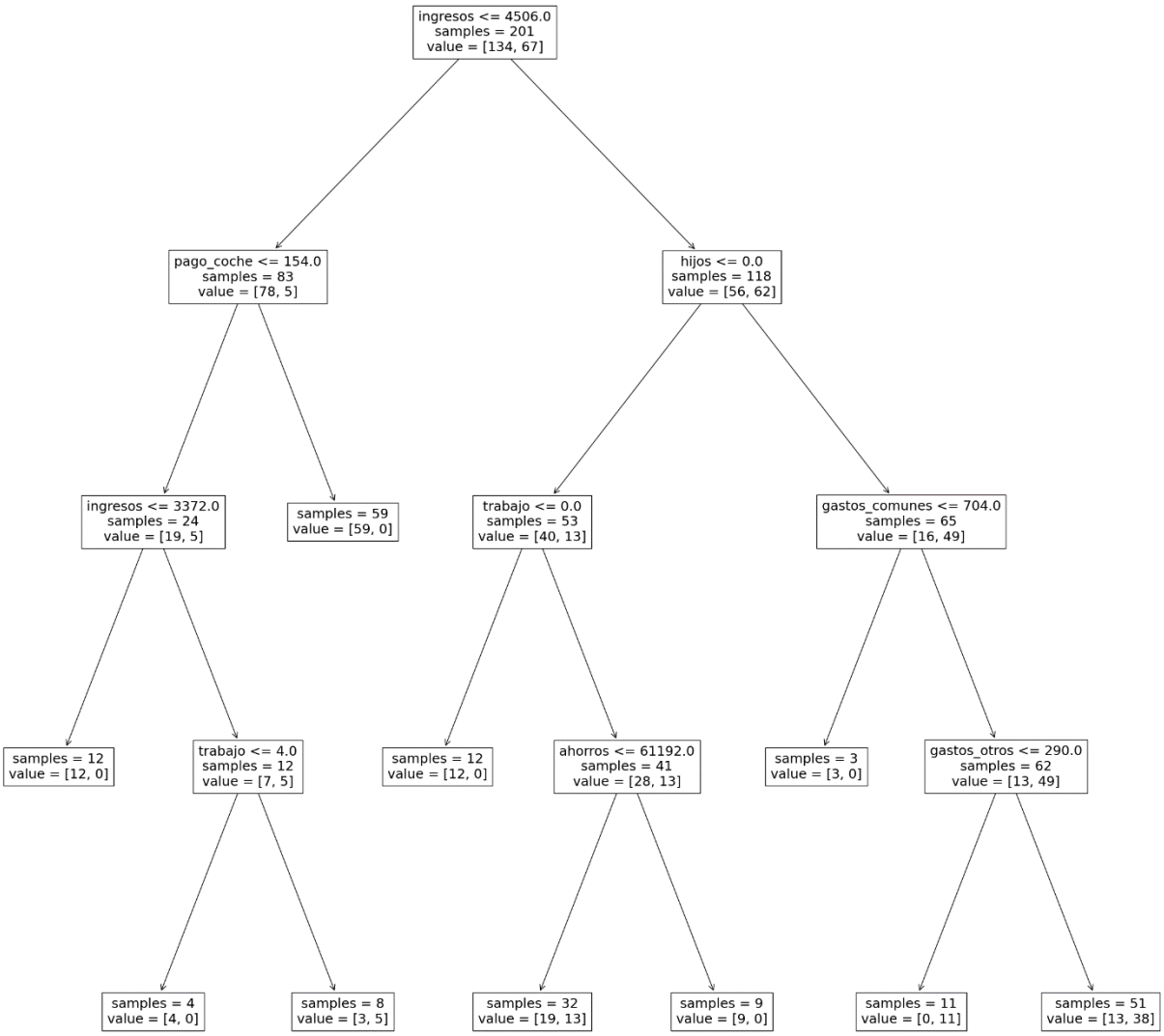
Busca en Internet información sobre estas cuestiones:

- A tu disposición tienes dos árboles de decisión: uno sobre compras y alquileres de una Vivienda, y otro sobre problemas de corazón:
  - En ‘datos’ aparecen los campos que se tomarán en cuenta a la hora de generar el árbol de decisión.
  - En ‘total_size’ es el porcentaje de los datos que se convertirán en parte de entrenamiento.
  - En ‘max_depth’ como profundidad máxima del árbol de decisión.
  - En ‘criterion’ como método para generar el árbol de decisión.
  - En ‘splitter’ para realizarlo de la mejor forma o de forma aleatoria.
- Cambia algunos de los valores en los campos anteriores y observa el resultado. Guarda varias imágenes de los cambios efectuados y de gráfico asociado.

En este primer ejemplo se puede ver el arbol de decisiones resultante de modificar el **test_size** a un valor de 1 y el **max_depth** a un valor de 4.

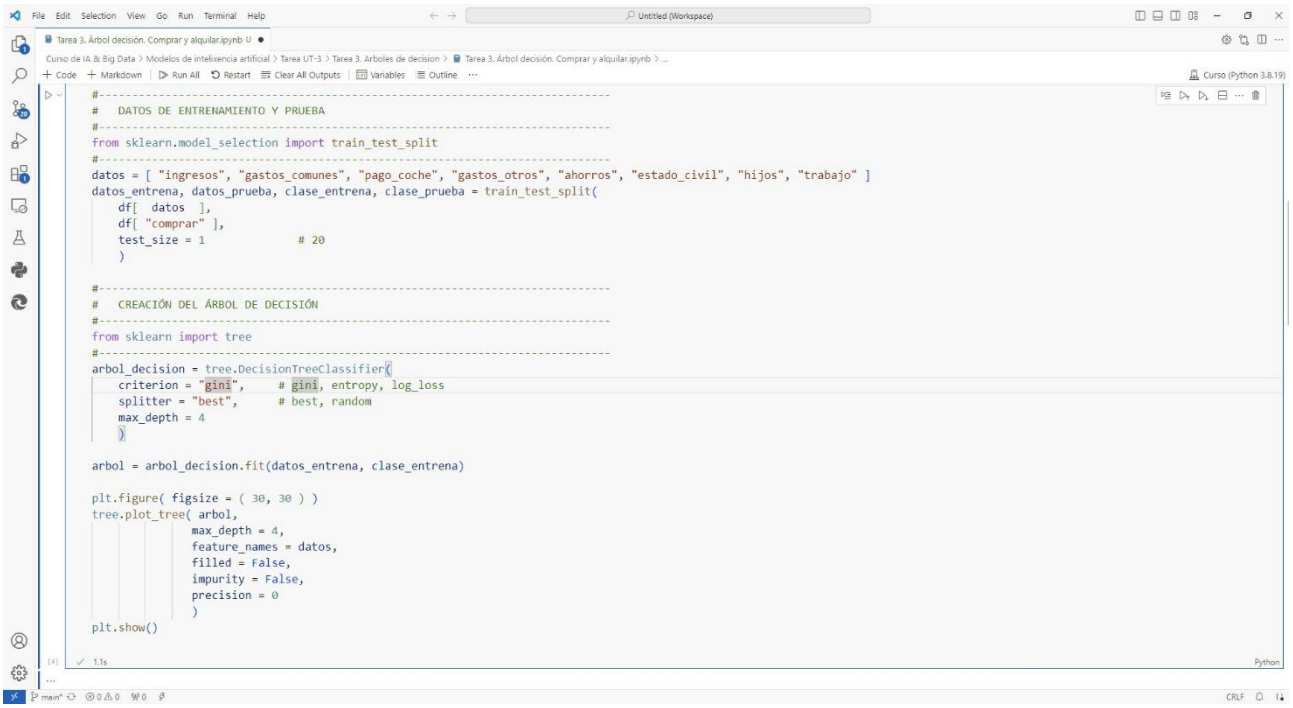


```
#-------------------------------------------------------------------------
#   DATOS DE ENTRENAMIENTO Y PRUEBA
#-------------------------------------------------------------------------
from sklearn.model_selection import train_test_split
#-------------------------------------------------------------------------
datos = [ "ingresos", "gastos_comunes", "pago_coche", "gastos_otros", "ahorros", "estado_civil", "hijos", "trabajo" ]
datos_entrena, datos_prueba, clase_entrena, clase_prueba = train_test_split(
    df[  datos  ],
    df[ "comprar" ],
    test_size = 1             # 20
    )

#-------------------------------------------------------------------------
#   CREACIÓN DEL ÁRBOL DE DECISIÓN
#-------------------------------------------------------------------------
from sklearn import tree
#-------------------------------------------------------------------------
arbol_decision = tree.DecisionTreeClassifier(
    criterion = "entropy",     # gini, entropy, log_loss
    splitter = "best",      # best, random
    max_depth = 4
    )

arbol = arbol_decision.fit(datos_entrena, clase_entrena)

plt.figure( figsize = ( 30, 30 ) )
tree.plot_tree( arbol,
                max_depth = 4,
                feature_names = datos,
                filled = False,
                impurity = False,
                precision = 0
                )
plt.show()
```


ingresos <= 4506.0
samples = 201
value = [134, 67]
pago_coche <= 154.0
samples = 83
value = [78, 5]
hijos <= 0.0
samples = 118
value = [56, 62]
ingresos <= 3372.0
samples = 24
value = [19, 5]
samples = 59
value = [59, 0]
trabajo <= 0.0
samples = 53
value = [40, 13]
gastos_comunes <= 704.0
samples = 65
value = [16, 49]
samples = 12
value = [12, 0]
trabajo <= 4.0
samples = 12
value = [7, 5]
samples = 12
value = [12, 0]
ahorros <= 61192.0
samples = 41
value = [28, 13]
samples = 3
value = [3, 0]
gastos_otros <= 290.0
samples = 62
value = [13, 49]
samples = 4
value = [4, 0]
samples = 8
value = [3, 5]
samples = 32
value = [19, 13]
samples = 9
value = [9, 0]
samples = 11
value = [0, 11]
samples = 51
value = [13, 38]

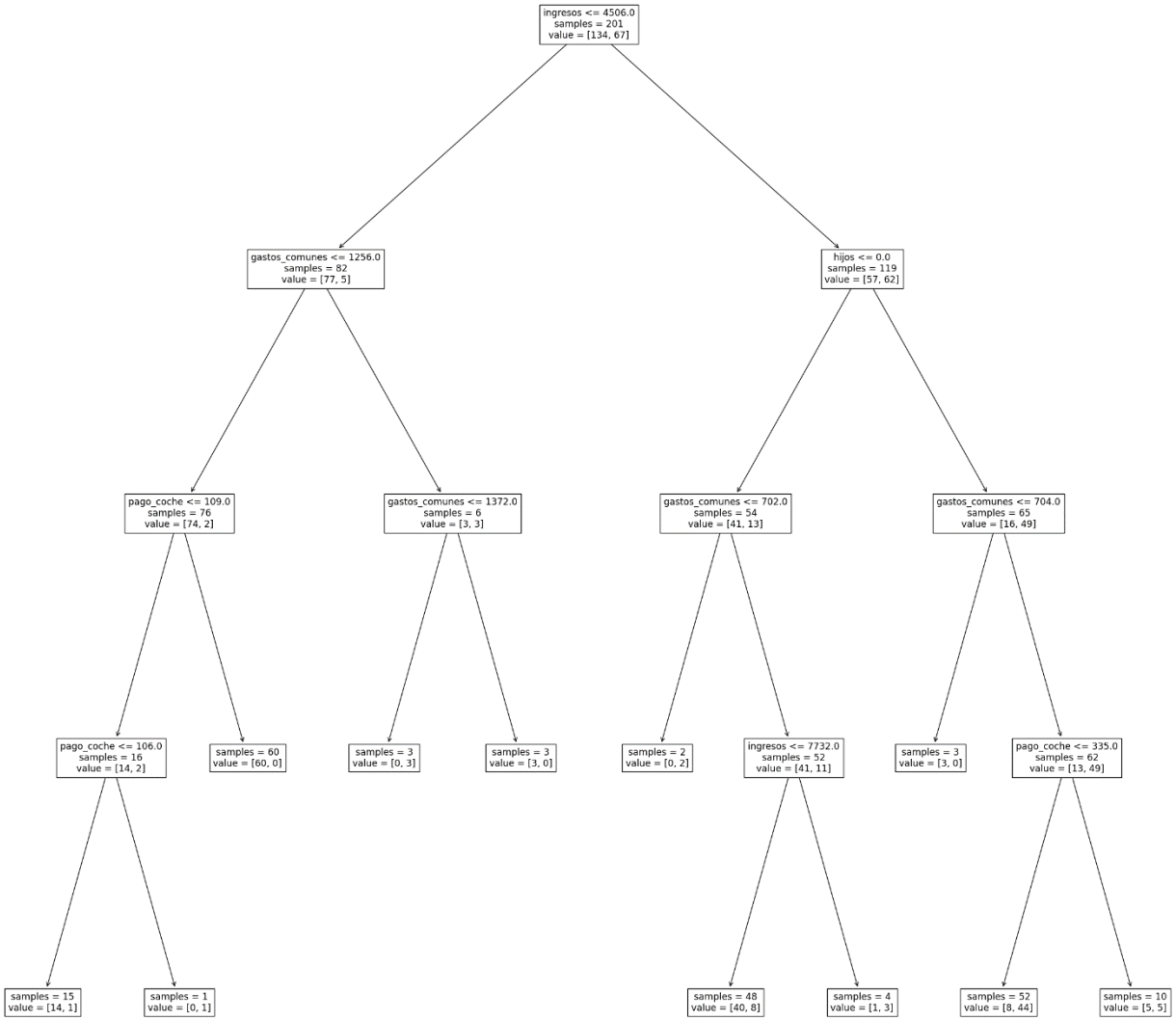
IES de Teis
Avda. de Galicia, 101
36216 – Vigo
886 12 04 64
ies.teis@edu.xunta.es

En este siguiente ejemplo se mantienen las modificaciones anteriores y se modifica el valor de **criterion** al valor “gini”.



```
#-------------------------------------------------------------------------
#   DATOS DE ENTRENAMIENTO Y PRUEBA
#-------------------------------------------------------------------------
from sklearn.model_selection import train_test_split
#-------------------------------------------------------------------------
datos = [ "ingresos", "gastos_comunes", "pago_coche", "gastos_otros", "ahorros", "estado_civil", "hijos", "trabajo" ]
datos_entrena, datos_prueba, clase_entrena, clase_prueba = train_test_split(
    df[  datos  ],
    df[ "comprar" ],
    test_size = 1             # 20
    )

#-------------------------------------------------------------------------
#   CREACIÓN DEL ÁRBOL DE DECISIÓN
#-------------------------------------------------------------------------
from sklearn import tree
#-------------------------------------------------------------------------
arbol_decision = tree.DecisionTreeClassifier(
    criterion = "gini",      # gini, entropy, log_loss
    splitter = "best",       # best, random
    max_depth = 4
    )

arbol = arbol_decision.fit(datos_entrena, clase_entrena)

plt.figure( figsize = ( 30, 30 ) )
tree.plot_tree( arbol,
                max_depth = 4,
                feature_names = datos,
                filled = False,
                impurity = False,
                precision = 0
                )
plt.show()
```

IES de Teis
Avda. de Galicia, 101
36216 – Vigo
886 12 04 64
ies.teis@edu.xunta.es



- Genera un árbol de decisión con alguna de las bases de datos incorporadas, o, con cualquier otra base de datos que tengas a tu disposición.

```
#-------------------------------------------------------------------------
#   INSTALAMOS LIBRERÍAS
#-------------------------------------------------------------------------
import pandas as pd
import matplotlib.pyplot as plt

#-------------------------------------------------------------------------
#   CARGAMOS LOS DATOS
#-------------------------------------------------------------------------
df = pd.read_csv( "winequality-red.csv" )
df
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

```
#-------------------------------------------------------------------------
#   DATOS DE ENTRENAMIENTO Y PRUEBA
#-------------------------------------------------------------------------
from sklearn.model_selection import train_test_split
#-------------------------------------------------------------------------
datos = [ "fixed acidity", "volatile acidity", "citric acid", "residual sugar", "chlorides", "free sulfur dioxide", "total sulfur dioxide", "density", "pH", "sulphates", "alcohol"
datos_entrena, datos_prueba, clase_entrena, clase_prueba = train_test_split(
    df[  datos  ],
    df[ "quality" ],
    test_size = 0.75              # 20
    )

#-------------------------------------------------------------------------
#   CREACIÓN DEL ÁRBOL DE DECISIÓN
#-------------------------------------------------------------------------
from sklearn import tree
#-------------------------------------------------------------------------
arbol_decision = tree.DecisionTreeClassifier(
    criterion = "entropy",      # gini, entropy, log_loss
    splitter = "best",      # best, random
    max_depth = 4
    )

arbol = arbol_decision.fit(datos_entrena, clase_entrena)

plt.figure( figsize = ( 30, 30 ) )
tree.plot_tree( arbol,
                max_depth = 4,
                feature_names = datos,
                filled = False,
                impurity = False,
                precision = 0
                )
plt.show()
```